Case-Based Reasoning for Simulation Based
Medical Specialist Intelligent Tutoring System

Final Report

Contract Number: M67004-93-C-0084

CDRL A003

Timothy Maher
Richard Stottler
Stottler Henke Associates, Inc. (SHAI)
Belmont, CA

May, 1994

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1994	Final

4. TITLE AND SUBTITLE

Case-Based Reasoning for Simulation Based
Medical Specialist Intelligent Tutoring System

5. FUNDING NUMBERS

C M67004-93-C0084

6. AUTHOR(S)

Timothy Maher
Richard Stottler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Stottler Henke Associates, Inc. (SHAI)

916 Holly Road
Belmont, CA 94002-2217

8. PERFORMING ORGANIZATION REPORT NUMBER

ARMY001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Institute
STRICOM
AMSTI-EI (Admiral Piper)
12350 Research Parkway
Orlando, FL 32826-3276

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

n/a

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; SBIR report,
distribution unlimited

12b. DISTRIBUTION CODE

DTIC QUALITY INSPECTED 2

13. ABSTRACT (Maximum 200 words)

In this Phase I SBIR research we proved the feasibility of a software solution to provide intelligent tutoring. Because students often learn best by example, we designed a tutoring system around example problems. Because Army training typically involves the use of simulations, we represent these example problems as scenarios which simulations utilize. The student performance with the simulation becomes his attempt at solving the example problem represented by the scenario. We applied the Artificial Intelligence (AI) technique of Case-Based Reasoning (CBR) to capture and represent the example problems (or cases) and provide explanations to students on problem solving techniques. The system is sensitive to a student's knowledge of problem solving principles and his performance and tailors teaching sessions accordingly. The body of knowledge contained in the system can be expanded over time simply through addition of more problems.

14. SUBJECT TERMS

Case-Based Reasoning, Intelligent Tutoring, Student
Diagnosis, Learning, Automatic Knowledge Acquisition,
Artificial Intelligence, Retrieval, Simulation

15. NUMBER OF PAGES

26

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

1.0 Abstract

In this Phase I SBIR research we proved the feasibility of a software solution to provide intelligent tutoring. Because students often learn best by example, we designed a tutoring system around example problems. Because Army training typically involves the use of simulations, we represent these example problems as scenarios which simulations utilize. The student performance with the simulation becomes his attempt at solving the example problem represented by the scenario. We applied the Artificial Intelligence (AI) technique of Case-Based Reasoning (CBR) to capture and represent the example problems (or cases) and provide explanations to students on problem solving techniques. The system is sensitive to a student's knowledge of problem solving principles and his performance and tailors teaching sessions accordingly. The body of knowledge contained in the system can be expanded over time simply through addition of more problems.

An intelligent tutoring system is an exciting supplement to traditional classroom instruction. It offers the possibility of individualized teaching and allows the student to work at his own pace, somewhat mitigating the problems of staff cutbacks and increasing complexity of subject area. In Phase I, we designed a generic tutoring system using a case-based reasoning paradigm. The feasibility of the system was proven through the implementation of a proof-of-concept prototype in the specific domain of Army Medical Specialist training.

Accesion For		
NTIS CRA&I		☒
DTIC TAB		☐
Unannounced		☐
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

1

2.0 Table of Contents

3.0 Background

The background section includes a description of the ITS problem, of Case-Based (Example-Based) Reasoning, and the Topic-Subtopic course representation strategy.

3.1 Problem Identification

Were the costs not prohibitive, students would greatly benefit from one-on-one instruction. A trainer could tailor a teaching strategy based upon his perception of the student's performance. Teaching is typically carried out through the presentation of examples and explanations and verified through testing on problems. The trainer could diagnose the individual student's weaknesses based on his errors and take tailored corrective action. An intelligent tutoring system could cost-effectively achieve the same benefits of one-on-one instruction.

Training devices and simulators are an effective tool in teaching information over a broad range of applications. A general ITS architecture which supports the incorporation of a simulator/simulation, either pre-existing or as part of the system, is such an effective tool. A case-based reasoning (CBR) ITS is a natural basis for such a general architecture since simulations operate with scenarios which are naturally represented as cases.

The goal of this research was to prove the feasibility of a software solution to provide intelligent training. Because students often learn best by example, we designed a tutoring system around example problems. Because Army training typically involves the use of simulations, we represent these example problems as scenarios which simulations utilize. The student performance with the simulation becomes his attempt at solving the example problem represented by the scenario. We applied the Artificial Intelligence (AI) technique of Case-Based Reasoning (CBR) to capture and represent the example problems (or cases) and provide explanations to students on problem solving techniques. The system is sensitive to a student's knowledge of problem solving principles and his performance and tailors teaching sessions accordingly. The body of knowledge contained in the system can be expanded over time simply through addition of more problems.

An intelligent tutoring system is an exciting supplement to traditional classroom instruction. An intelligent tutoring system offers the possibility of individualized teaching and remedial teaching, allowing the student to work at his own pace, somewhat mitigating the problems of staff cutbacks and increasing complexity of subject area. Additionally, training instructors could encourage students to enter their own problems into the

· system (with supervision), making the system more relevant to the particular training class.

The potential commercial applications of an intelligent tutoring tool are enormous. All companies or government agencies involved in training could benefit from the reduced cost and increased quality and consistency of the education. Because we envision a general-purpose CBR intelligent tutoring system stemming from this effort, little software developer time will be needed to customize it for particular domains.

3.2 Technical Background

Many studies have been performed on utilizing prior experience, or analogical reasoning, in various domains and on representing prior situational knowledge. Humans reason about a given situation based on knowledge about that situation and associations to previous experiences. This same reasoning process applies to learning.

Case-Based Reasoning (CBR) is the field of AI which deals with the method of solving a current problem by retrieving the solution to a previous similar problem and altering that solution to meet current needs. Case-based reasoning is a knowledge representation and control methodology which can assist planners and designers in making complex, domain-specific decisions, designs, or problem assessments and recommendations based upon previous experiences and patterns of previous experiences. These previous experiences, or "cases" of domain-specific knowledge and action, are used in comparison with new situations or problems. These past methods of solution provide expertise for use with new situations or problems. The general problem of teaching students is well suited for the application of such a case-based reasoning method.

CBR systems offer enormous benefits compared to standard AI approaches. The knowledge elicitation bottleneck is largely circumvented. Cases can be automatically acquired directly from domain experts. Rules or other models of expert knowledge always require the intervention of a knowledge engineer. Instead of having to elicit all the knowledge required to derive a solution from scratch, only the knowledge required to represent a solution is needed.

The fact that the knowledge elicitation bottleneck is largely circumvented is especially obvious with simulation-based intelligent tutoring systems. The standard ITS approach starts with modeling the expert in the chosen domain. An ITS is then constructed migrates the student's

mental model toward that of the expert. This standard
approach requires two time-consuming domain dependent steps
which require costly knowledge engineering. The costs
associated with this approach are impractical. Furthermore,
it does not make explicit use of the simulations already
used for training in many domains.

In stark contrast, consider the CBR approach to ITS
development. The first step is to develop a case base of
examples. Typically a knowledge engineer is required to
help define the form or structure of the cases but domain
experts (or clerical staff working under their direction)
can input the information of each case. It is very natural
for these cases to take the form of scenarios used by
training simulations, thus making use of the existing
resource to help define and structure the domain knowledge.
These cases typically reference principles or other domain
knowledge which the student must learn. The domain expert
or software engineer must simply provide media which
describes this domain knowledge. Because much less
knowledge engineering time is required, the costs associated
with the CBR approach are often an order of magnitude
smaller.

The student model very naturally drops out of this
representation as the set of scenarios the student has seen
as well as his performance on them. From this performance
it is straight-forward to derive the domain knowledge
mastered by the student and which knowledge still needs to
be conveyed. This knowledge is conveyed by showing the
student the relevant information and scenarios. The method
of teaching the student is therefore domain independent,
thus leading to additional considerable savings in ITS
development costs.

Conventional knowledge base technology dictates a
single, fixed problem solving methodology. With CBR, each
case, in the extreme, can represent a different methodology.
Therefore, many problem solving methodologies are
represented and, since new cases are continually added
automatically, a CBR system's problem solving methodologies
can change with time, thus improving its performance and
staying up to date and relevant automatically.

Intelligent tutoring is a natural application for CBR.
Students learn best from examples. These examples consist
of a problem, its solution, and an explanation as to how
that solution was derived. This system seeks to aid that
natural process through the use of a CBR system to document
examples and the principles and procedures needed to solve
them. Teaching will be accomplished through the
presentation of appropriate examples. Testing is performed
by presenting the problem part of an example and comparing
the student's solution to the stored solution.

3.3 Electronic Course Organization

A typical course at the is divided into sections.
These sections are further divided into subsections, and so
on. This hierarchical representation allows the large
amount of information to be organized in a comprehensible
way. An ITS for the course should parallel this structure.
In our ITS we call the basic organizational unit a topic. A
course is represented by a specific topic. This topic
includes a list of subtopics which correspond to sections.
Those subtopics include subsubtopics corresponding to
subsections, and so on. Each topic contains an identical
structure which includes the list of subtopics, the list of
relevant examples, the list of multimedia descriptions of
the topic, and the principles which the topic illustrates.
The fact that each topic is identical in structure
facilitates student learning of how to interact with the
system. Once he understands how to interact with one topic,
he understands how to interact with them all.

4.0 Methodology

This section describes the tasks which were performed as part of this Phase I effort and the technical accomplishments. At the completion of Phase I, all technical objectives were achieved. The Phase I technical objectives were:

1. Establish a Pertinent Specific Domain: Working closely with Army representatives, the Medical Specialist domain was chosen as the problem domain. Medical Specialist training was considered an area of very high priority. The first course in the 91 B sequence was investigated but a computer based training system was already under development in a separate effort. We obtained the most critical tasks taught in the second course in the sequence and this became our primary domain.

2. Knowledge Elicitation: We captured and operationalized a sample of the expertise of medical specialists/teachers in our critical tasks.

3. Case-Based Representation and Reasoning Architecture: We produced a generic architecture for a case-based reasoning ITS.

4. System Integration With Existing Environment: We established that there were no requirements for the prototype to integrate with existing simulations, hardware, data bases, or domain specific tools. We decided to create our own simulations for the prototype.

5. Prototype Development: We developed a proof-of-concept prototype which demonstrated important CBR functionality within a subset of the Medical Specialist domain.

An eight task approach was proposed for developing the techniques and implementations required for the objectives. The tasks were:

1. Develop the specific trial problem domain for a CBR trial application.
2. Define the Preliminary Case Structure for the elicitation procedure.
3. Conduct application of CBR elicitation for the purpose of developing case histories.
4. Develop/Design the structure for the case base.
5. Develop/Design the retrieval process and reasoning structure for the system.
6. Implement the Prototype
7. Assess the validity of the prototype
8. Prepare the Phase II design and final report.

Each task is described in more detail below:

1.   Develop the specific trial problem domain for a
CBR trial application.

Working closely with Army representatives, the Medical
Specialist domain was chosen as the problem domain. Medical
Specialist training was considered an area of very high
priority. The first course in the 91 B sequence was
investigated but a computer based training system was
already under development in a separate effort. We obtained
the most critical tasks taught in the second course in the
sequence and this became our primary domain.

2.   Define the Preliminary Case Structure for the
elicitation procedure.

Working with medical specialists, we determined a
preliminary representation for our two types of cases: EKG
and patient simulations. The cases are presented as
simulations and consist of attributes relating to the
simulated scenario as well as problem-solving principles and
methods. We also begin examination of potential similarity
metrics and retrieval methods.

3.   Conduct application of CBR elicitation for the
purpose of developing case histories.

In cooperation with our local paramedic domain experts,
we elicited cases and other knowledge required for the CBR
ITS.

4.   Develop/Design the structure for the case base.

Once the knowledge was collected a refined structure
for representation was developed.   An object-oriented
approach was used to represent a case. Not only was the
case structure needed for the examples and tests but also
the structure for the overall case library. The case
structure considered the various retrieval processes to be
used.

In addition to refining the preliminary case structure
for the trial domain, the general, domain-independent case
structure was also refined so that the trial domain remained
a specific instance of the general case structure.

5.   Develop/Design the retrieval process and reasoning
structure for the system.

We examined potential similarity metrics and retrieval
methods. Similarity is based primarily on principles.
Examples are similar if they illustrate the same principles.
The principles in the prototype are the indications and

contraindications of our critical tasks and the procedural steps of these critical tasks. In addition, recognition of each dysrhythmia is also a principle.

In addition to the retrieval process, the EKG and patient simulations were designed. The EKG simulation presented the student with the picture of an EKG strip, and the student must correctly decide which dysrhythmia the EKG represented. The patient simulation begins with a description of the scene and the patient. Then an animation of the patient is presented, and the student is able to perform various actions on the patient, such as checking the pulse, performing a quick-look EKG, performing intubation, etc.. The simulation models the condition of the patient through time based on his condition and the student's actions. Once the student has mastered the steps of the procedure, the simulation will automatically perform the steps of the procedure when the student requests to perform the procedure. The student will no longer have to specify each step of the procedure.

In the patient simulations, actions can be specified from a main menu or by clicking on the appropriate part of the simulated patient's body where the action is normally performed. For example, the pulse may be checked by clicking on the wrist or neck of the simulated patient or by selecting "Check Pulse" on the main menu of the simulation. For the critical tasks, such as intubation, the student must master the steps of the procedure by requesting that each action be performed on the simulated patient in the correct sequence. As actions are performed on the patient, the results for the "check" actions are displayed in text to the student. If the action requires an audible response, an audio is also played for the response. When actions result in physical objects being placed on the patient, the objects are shown in the simulation via animations. For example, when the patient is ventilated with oxygen, a moving bag valve mask is displayed over the patient's mouth.

The list of examples and the list of tests available to a student within a specific topic are automatically generated at the time the student requests to run a test or to see an example. Case-based reasoning is used to find the best set of tests or examples for the student at that time based upon the principles the student has mastered, the principles in the topic, the examples and tests already seen by the student, and the principles in the examples and tests.

Additionally after seeing an example, the student may request a similar example, and after failing a test, the student may request a remedial example. Case-based reasoning is also used to generate the list of examples for these requests.

One of the factors used in the similarity search used to retrieve examples and tests is the usefulness of the example or test. All cases start with the same usefulness. After running an example or test, the student can provide feedback on that particular case. If the student provides negative feedback, the case will be less similar than an otherwise comparable case and will be lower in the list of examples or tests retrieved in the future. If the student provides positive feedback on the case, the case will have a higher similarity score and will be placed higher in the list of retrieved examples or tests.

In addition to the retrieval and simulation reasoning, the remediation process was also designed. After running a test, a student may request remediation in one of three ways. In the first, the student can request a remedial exercise. A remedial exercise is found through a similarity search of the cases using the principles which the student has failed. In another form of remediation, the student may request a remedial example. A similarity search is also performed to find the list of appropriate remedial examples. The third form of remediation is through the presentation of remedial topic information. The student can request remediation on one of the principles which he has failed. The student is then presented with more detailed information on the topic, and if the topic contains a question, the question is directed toward the deficiency of the student in his last test. For example, if the student fails the intubation procedure by performing the steps in the incorrect order and requests remediation on the intubation principle, he will be presented with more detailed information on the intubation procedure and will then be presented with a question on the exact step of the intubation procedure which he failed.

6.   Implement the Prototype

Based on the previous tasks, we developed a small proof-of-concept prototype. This prototype provides a sample of the "look and feel" of the system and contains representative CBR functionality that operates on a subset of the intelligent tutoring domain. It demonstrates how a trainee can learn via examples and interactive multimedia and how the system can automatically modify its course of instruction to meet current student needs.

7.   Assess the validity of the prototype

To prove the feasibility of this effort, the validity of the prototype was evaluated. The prototype was demonstrated to instructors and training developers of the 91B course at Ft. Sam Houston and to representatives from the ARPA support team at Camp Dodge. Modifications to the

prototype were made based upon feedback from these
demonstrations.

       8.   Prepare the Phase II design and final report.

      The final report describes the development and
architecture of both the general and the specific case
structures and retrieval methods and includes the Phase II
design. This design includes the architecture for all
modules. The evaluation of the prototype in its trial
domain is presented. A future research section outlines the
requirements needed to develop a general intelligent
tutoring system, applicable across many subjects. This
general system would serve as a starting point for a
commercial system.

5.0 Prototype Description

The prototype developed for the Phase I effort served as a successful proof-of-concept illustrating the feasibility of a software solution to providing intelligent, interactive, multimedia tutoring. There are two main perspectives from which the prototype can be described:

1. General Tutorial Structure and Techniques
2. Medical Specialist Course Topics

The Phase I effort resulted in the design of a general structure for presenting domain information in a tutorial, and the implementation of a number of innovative techniques for the manipulation and presentation of this information in the course. The ITS prototype essentially consists of the application of this tutorial structure and the accompanying techniques to the Medical Specialist domain.

5.1 General Tutorial Structure and Techniques

The general tutorial system produced for the Phase I effort consists of two primary components: (1) the structure for the student information and the course materials through which the students pass, and (2) the manipulative techniques applied to the information in this structure and the students who use it, which make the system innovative, effective, and more useful than an *on-line textbook*.

5.1.1 Tutorial Structure

The structure of the tutorial consists of the model for organizing the course information, the model for presenting the course information, and the model for monitoring and updating the student information.

5.1.1.1 Organizational Hierarchy

Course materials in the prototype are organized and automatically presented in a hierarchical fashion. A course is made up of topics. A topic is made up of subtopics. A subtopic may be composed of subsubtopics and so on. The depth is arbitrary and can vary between courses or for different parts of the same course.

In an object oriented system, such as the prototype, real world objects or concepts are mapped onto computer representations called objects. These objects can hold data which might include references to other objects. The objects and references can be thought of in a visual way as a network. If the references are hierarchical, as in the topic-subtopic relationship, the structure is called a tree.

The hierarchical structure for a course can be represented as a tree structure. The root node of the tree represents the course. The root node's children represent topics, and their children nodes represent subtopics. We call each node a Topic which can represent either the node for the entire course (the root node) or any of its constituent parts. A Topic contains a multimedia description and a possibly empty list of its children. Progression through the course involves moving up and down the hierarchy of Topics and leaf Topics. After viewing the description, the student would then begin interacting with each of the root's children (course topics), and their children, in turn.

Topics that do have subtopics play a categorical role, dividing large segments of information into smaller chunks for easier digestion. Topics with no subtopics are considered leaf Topics. A leaf Topic's description contains the basic course material which needs to be conveyed to the student. Associated with each Topic is a list of principles illustrated by that topic. From these principles, the system derives interactive examples that illustrate the Topic. A part of interacting with the Topic is interacting with one or more of the examples. From the topic's principles, the system also derives a list of examples which can be used for testing.

5.1.1.2   Topic Presentation

Descriptions

Each topic may contain several descriptive presentations in most any media – text, image, hypertext, video, audio, etc. These can serve two functions: (1) to give the student the initial information relevant to the topic, and (2) to make the overall course and the specific topic more interesting and even entertaining.

Topics may contain multiple sequences of media. One sequence is the default sequence shown initially. If the student has failed a principle within the topic, a second, more detailed sequence is shown.

Additionally, topics may contain multiple choice questions. The questions can be presented with other media, such as an image. The student then selects an answer, and the system displays the result, indicating the right answer if it wasn't chosen. Questions within a topic are not fixed. A set of questions is defined to be shown in a specific place within a topic, and the system randomly selects one of the set to display. If the student has failed a specific part of a principle, such as a step within a critical task procedure, the question addressing the specific step is shown.

Examples

Because students often learn best by example, the
tutoring system is designed around example simulations. The
examples are made more vivid by interactive multimedia
technology. An example consists of a problem description,
solution, and explanation or steps leading to the solution.
Each section of the example can consist of several elements
each of which might be of any media type.

For example, in the Medical Specialist ITS prototype,
the example patient simulations consist of a description of
a patient and the set of actions which are appropriate for
that patient. The patient information is used to produce a
graphical representation of the patient. The prototype
automatically simulates and animates the patient as the user
performs actions, such as ventilating the patient, checking
the pulse, or defibrillating the patient.

Examples can be positive or negative. Positive
examples show correct solutions, while negative examples
show incorrect solutions or problem-solving methods.
Negative examples generally include an explanation as to why
they are wrong and a demonstration of the undesirable
results. Both of these kinds of examples can be quite
useful for the student, providing concrete illustration of
correct and incorrect responses, and the results they
produce.

The ITS prototype was designed for generality, both in
the format of the examples used and the overall presentation
of the course. This serves to illustrate the wide range of
options available when tutoring is carried out on a computer
platform. Different kinds of examples can be presented in
different media for both passive and interactive student
participation, making the tutoring more dynamic. So in the
examples described above, the situation descriptions and
solution explanations can be presented (like everything
else) in a variety of media.

In addition to the ability to present examples in
different ways, this prototype can present different kinds
of examples. For example, the Defibrillation topic uses
examples which are completely different from the examples
used for the EKG topics. These examples deal with simulated
patients with heart conditions requiring defibrillation or
heart conditions very similar to those requiring
defibrillation. Various actions can be performed on the
simulated patient, such as doing a Quick Look of the EKG of
the patient. Beside the role that this topic and its
associated examples play in the overall subject matter of
the course, they reveal the adaptability of a computer-based

14

tutoring system to different illustration techniques and
formats.

Exercises

Among the most significant elements of a course are the
exercises or tests used to measure students' understanding
and progress. Each topic may have several associated
principles, from which tests are derived. In the prototype,
tests essentially are examples, with one difference in
presentation -- the student is not allowed access to the
example's solution, but is instead required to produce the
solution. The following would be a typical sequence of
events:

1.   The student enters the Intubation topic,   eps
through the topic descriptions, and then views the
description, simulation, and explanation for an example.

2.   The student then clicks on the "Run Test" button,
which presents a list of tests. The student selects one of
the tests to run.

3.   The description for the simulated patient of the
selected test is displayed. The student views this, and
clicks on the Start button, which brings up the animation of
the patient.   The idea is that since the student has
already seen an example of a similar patient scenario,
determining the condition of the test patient and the
appropriate actions for this condition should be
straightforward.

4.   The student interacts with the animation to
determine the patient condition and perform the correct
actions.

5.   After passing or failing, the student is returned
to the simulation start/finish window, and may view the
explanation for the test's solution, which may differ from
his own solution.

Applying examples as tests is another way that a
computer-based tutoring system can be dynamic, allowing for
easy revision of test formats as well as variety among
existing tests. Also, the student's solution and solution
steps can be compared to that of the exercise for grading,
immediate feedback, and remediation.

5.1.1.3   Student Records

In order to tailor the course of study to the
individual student, the system keeps an explicit model of
each student. The student model includes a list of the
topics, examples, and exercises the student has examined and

experienced, along with performance information where applicable. Additionally, the principles mastered and failed by the student are recorded. The results of the student on various exercises is also tracked. This information is useful for keeping track of the student's progress and performance in the course. With this information available for each student, the system can produce recommendations for the student to follow which may facilitate the effectiveness of both the student and the system.

Based on the pattern of the student's unsatisfactory performances, a set of topics and principles, or combinations of them, can be developed which form a hypothesis as to what information the student does not understand. Based on this hypothesis and the exercises failed, similar examples can be shown to increase the student's understanding. Additionally, topic information can be tailored to address the deficiencies of the student. A new set of exercises can also be generated for testing the success of the remediation.

The student model can also be referenced by the course instructor to monitor the student's progress through topics and his performance on the problems.

5.1.2   System techniques

There are a host of strategies used in the Medical Specialist ITS prototype that increase the benefits it brings to the application of computers to tutoring. These include:

1. Similar example retrieval
2. Multimedia presentations
3. "Real world" simulations
4. Dynamic course progression
5. Object-oriented course material representation

5.1.2.1   Similar Example Retrieval

The most significant innovation is the use of Case-Based Reasoning (CBR) techniques for similar example retrieval. CBR is the field of Artificial Intelligence (AI) which deals with the method of solving a current problem by retrieving the solution to a previous similar problem and altering that solution to meet the current needs. CBR is well-suited for a tutorial system based on examples, because it can be used to suggest directions for the student to pursue. This may involve providing the student with an example that is similar to one he selects, or suggesting remedial examples or exercises to help the student with specific failed exercises.

Retrieving similar examples is important for showing the student more angles on the same problem and formulating a set of remedial examples and exercises. One important aspect of similarity in this project is the concept of principles. Our notion of principles is a general concept. A principle might be a diagram or animation in addition to a sentence, paragraph or whole Topic. The principles are what it is important for the student to learn and each has a name or some unique way to identify it, such as a diagram's file name.

Examples are similar if they illustrate the same principles. This principles list can be entered for each example or automatically derived based on the principles referenced in the text of the description, explanation, or solution steps. Similarity can also be based on the text or other referenced media used in the problem description, solution, explanation, or explanation steps. The topics that examples are associated with can also be used for similarity. Examples associated with the same subtopic are similar. Examples associated with subtopics that share a common parent are also similar, but less so.

There are several main uses in the prototype for this similarity retrieval capability on the examples. The first is directly by user request. Having examined all the examples provided with a topic, the student may be interested in seeing if there are any further similar examples. In this case, the student may get this list simply by clicking on the "Get Similar Example" button and selecting one of the examples displayed.

The second use for the similarity retrieval capability involves the provision of remedial examples or exercises for the student based on the principles he has not understood (which are determined from the student's failed exercises). For a given failed exercise, the system can produce either a list of similar examples which the student has not yet seen (to help the student understand the principles), or a list of similar exercises which the student has not yet tried (to test the success of the remedial examples). These remedial examples and exercises are in addition to the remedial topic information that the student may request. Both lists are ordered in terms of similarity to the failed exercise.

The third use for the similarity retrieval is dynamic computation of relevant examples and exercises. There are no static lists of examples and exercises associated with each topic. Rather, when the student enters a topic, examples and exercises are dynamically retrieved that illustrate principles described by the topic and do not use principles which the student has not yet learned. This dynamic computation makes the system much more tolerant to differing sequences preferred by different students and

facilitates case addition. A domain expert entering a new case does not need to specify for which topics the case can be used. This is calculated automatically.

The prototype allows students feedback to affect the similarity retrieval. Students can encourage or discourage retrieval of cases by giving the degree of usefulness of the cases.

5.1.2.2 Multimedia Presentations

Interactive multimedia is used to present both the course material and the examples. The multimedia capabilities supported by the prototype include:

- Interactive Simulations
- Interactive Animations
- Video
- Audio
- Hypertext
- Photos
- Drawings

Interactive simulations are especially useful as examples for a course. For example, in the Phase I prototype for the Medical Specialist course, a simulation is used to show the effect of performing actions on a patient. The student can vary the actions performed and rerun the simulation. This effectively allows one example to behave as many and allows students to ask what-if questions. Rerunning the simulation allows an immediate answer to the questions.

Another technique, related to simulations are interactive animations. In the example above, instead of just printing the results of each action, the result of performing an action can be animated. Seeing the steps of the intubation procedure animated as the student chooses each action makes the examples and exercises much more vivid and therefore more likely to be remembered.

Video can be especially helpful in the descriptive or motivational section of each topic. Videos can be made interactive by allowing user controlled slow motion, freeze frame, rewind, fast-forward, and branching. Branching is allowed at certain points in the video by giving the user a choice about which video among a set of choices to see.

5.1.2.3 "Real-world" Simulations

The simulations presented in the tutorial produce results that reflect the actual outcomes that would be expected based on the defined conditions. The objects in the simulations (e.g. the patient condition) are defined by

their behavior and act appropriately according to these
specifications.

This brings enormous advantages to a tutorial system,
because it yields immediate results for the student. When
the student can test his own performance on a new scenario,
he does not need to wait for the evaluation of a course
instructor to be certain of his success or failure.

5.1.2.4   Dynamic Course Progression

A standard progression is outlined for the student to
follow through the course. This order is suggested by the
course outline and involves an initial ordering for his
traversal of the topics and subtopics which comprise the
course material. The student is free to follow his own path
through the course. The ordering could be easily be fixed
to allow the course instructor to specify the required order
of the topics.

5.1.2.5   Object-oriented Course Material Representation

By using an object-oriented design method, an object
hierarchy is created which closely matches the real world
objects and concepts. This is especially useful in tutorial
applications, because these concepts are what need to be
communicated to the student. If the structure of the
tutorial itself resembles that of the course material, this
contributes to the clarity of the course. There are also
many programming advantages to an object-oriented approach,
including adaptability, the ease and speed of information
access, and the modularity of the code.

5.2   Medical Specialist Course

The following critical tasks from the Soldier's
Training Manual and Trainer's guide were recommended by the
Army to be covered by our ITS:

    Prepare Intubation Equipment
    Intubate a Patient
    Extubate a Patient
    Perform a Needle Cricothyroidotomy
    Perform a Surgical Cricothyroidotomy
    Perform Needle Chest Decompression
    Manage Cardiac Arrest

Working with our domain experts, we developed the
following outline for the topics in our Medical Specialist
ITS:

    Advanced Airway Management
        Anatomy and Physiology of the Respiratory System

Assessment of the Airway and Breathing/Ventilation
Review of Basic Airway Management
Advanced Airway Management Skills
    Airway Obstruction Removal
    Endotracheal Intubation
    Cricothryroidotomy

Chest Trauma
    Thoracic Anatomy and Physiology
    Assessment of Thoracic Injuries
    Management of Life-Threatening Thoracic Injuries
            during Primary Survey
        Injuries that Jeopardize the Airway
        Injuries that Jeopardize Breathing
            Open Pneumothorax
            Tension Pneumothorax
            Flail Chest
        Injuries that Jeopardize Circulation
            Massive Hemothorax
            Cardiac Tamponade

Cardiovascular Emergencies
    Anatomy and Physiology of the Cardiovascular
        System
    Patient Assessment
    Cardiac Dysrhythmias
        Basic Concepts of EKG Monitoring
        Specific Cardiac Dysrhythmias
    Management of Cardiac Emergencies
        Review of Basic Life Support
        Advanced Life Support
        Manual Defibrillation
        Cardioversion
        Precordial Thump

5.2.1   Advanced Airway Management

The Advanced Airway Management topic covers, in detail,
the critical tasks of preparing the intubation equipment,
intubating a patient, and extubating a patient. To support
the discussion of these tasks, a review of the anatomy and
physiology of the respiratory system is included, as is the
assessment of the airway and breathing and a review of basic
airway management. Within the intubation topic, the
examples and tests presented include simulations of patients
with intubation indications and patients with intubation
contraindications. If other principles and tasks, such as
defibrillation or needle chest decompression, have been
mastered, simulations involving those tasks are also
available to the student as tests and examples.

5.2.2   Chest Trauma

The Chest Trauma topic covers, in detail, the critical task of needle chest decompression. To support the discussion of this task, a review of the anatomy and physiology of the thorax is included, as is the assessment of thoracic injuries. Within the tension pneumothorax topic, the examples and tests presented include patient simulations which require needle chest decompression and simulations which contain contraindications for needle chest decompression. If other principles and tasks, such as intubation or defibrillation, have been mastered, simulations involving those tasks are also available to the student as tests and examples. Additionally, other injuries which present similar symptoms but different treatments, such as cardiac tamponade, are also presented.

5.2.3 Cardiovascular Emergencies

The Cardiovascular Emergencies topic emphasizes two major skills, recognition of cardiac dysrhythmias and management of cardiovascular emergencies. To support the discussion of these skills, a review of the anatomy and physiology of the cardiovascular system is included, as is the assessment of patients with cardiovascular emergencies.

5.2.3.1 Dysrhythmias

The dysrhythmias topic presents, in detail, the recognition of specific cardiac dysrhythmias. The basic concepts of EKG monitoring are presented. The specific cardiac dysrhythmias are also covered in detail. Each dysrhythmia is included in its own subtopic. Within each subtopic, the student is presented with a question to reinforce the dysrhythmia recognition. Additionally, tests on recognition are available within the dysrhythmia topic.

5.2.3.2 Management of Cardiovascular Emergencies

The Management of Cardiovascular Emergencies topic covers, in detail, the critical task of defibrillation. To support the discussion of this task, a review of basic life support and a topic on advanced life support are included. Within the manual defibrillation topic, the examples and tests presented include patient simulations which have indications or contraindications for defibrillation. Additionally, if other principles and tasks, such as needle chest decompression, have been mastered, simulations involving those tasks are also available to the student as tests and examples.

6.0 Phase II Design

The ITS can be developed to run under DOS/Windows, UNIX or both. PC platforms are probably the most appropriate, although others can be accommodated. The ITS consists of a

generic ITS development capability (GITSDC) and individual
ITSs built using that capability.

6.1   Generic ITS Development Capability

SHAI has significant experience in developing generic
development capabilities. Both the PROSPER and ESTEEM
products where tools to aid in the development of CBR
systems in different domains. Our planning and scheduling
tool developed for NASA, AMP, can be easily customized by
users to operate in any of several domains.

6.1.1   Use of Examples

The ITS is designed around example problems and
simulations. An example consists of a problem description,
solution, and explanation or steps leading to the solution.
Each section of the example can consist of several elements
each of which might be of any media type. The problem
description might consist of hypertext and animations
stating and showing the initial condition of the simulated
patient. The solution might be the correct set of actions
for the simulated patient along with descriptions explaining
why a specific action is appropriate. An animation
capability can vividly illustrate the results. The
explanation or solution steps might consist of a video or
audio clip along with annotated figures.

An exercise can be extracted from an example by only
showing the student the problem description. He must then
generate the solution himself. Based on both the simulation
results and comparison, the exercise can be graded. The
student then receives immediate feedback both in the form of
simulations showing how well a solution performs and with a
presentation of the correct answer along with explanations
or solution steps. The student can request similar examples
or related course material to aid in his understanding.
Similar examples are different examples which illustrate the
same principles. This gives the student a different angle
and broader base of understanding for the principles.

In the case of erroneous responses by the student, the
ITS can design a remedial course of study consisting of new
examples illustrating the principles that the student fails
to understand. This list of difficult principles can be
generated from the exercises that the student has failed.
Additional course material explaining the principles can
also be used. At the end of the remedial course exercises
are selected to test the student's grasp of the difficult
principles. The exercises are extracted from similar
examples which the student has not yet seen.

Student responses on exercises can be automatically
graded in a number of ways. The student's solution can be

simulated and the results compared to specified results for the example. In domains with a small number of correct answers, the student's solution can be compared to the set of correct ones. If the solution process involves a number of steps entered by the student, these steps can be compared with the correct solution steps for the exercise. The exercise can also store some simple information to be used for grading. Are extra steps OK? Can the steps be out of order or skipped? Is there a set of definitely incorrect steps? These steps may be represented and displayed in a number of ways. They may be simple text input by the user, or a series of forms or other media. Incorrect steps in the student's solution can be found and highlighted along with an explanation for the student.

6.1.2   Similarity

Retrieving similar examples is important for showing the student more angles on the same problem and formulating a set of remedial examples and exercises. SHAI has enormous experience in developing similarity retrieval techniques.

One important aspect of similarity in this project is the concept of principles. Our notion of principles is a general concept. A principle might be a diagram or animation in addition to a sentence, paragraph or whole Topic. The principles are what it is important for the student to learn and each has a name or some unique way to identify it, such as a diagram's file name.

Examples are similar if they illustrate the same principles. This principles list can be entered for each example or automatically derived based on the principles referenced in the text of the description, explanation, or solution steps. Similarity can also be based on the text or other referenced media used in the problem description, solution, explanation, or explanation steps. Which topic the example is associated with can be also used for similarity. Examples associated with the same subtopic are similar. Examples associated with the subtopic which share a common parent are also similar, but less so.

6.1.3   Representations

The Topic representation parallels the hierarchical (tree) structure of the course. Each Topic contains a slot which holds the list of subtopics. If the list is empty, it indicates that the topic is a leaf node. The system allows students to navigate through the tree by presenting this list of subtopics to the student as buttons. Selecting one opens that topic. Closing a subtopic leaves the student at the previous topic.

The Topic contains a slot which holds the description or subject material for a topic. That slot contains a list of the multimedia objects which are displayed when the topic is opened. Multimedia objects are described below.

The topic also contains a list of the interactive examples that the student should experience. These can be mandatory or optional. A facility exists within each topic to retrieve similar topics for viewing by the student. The topic contains a list of the examples to be used as exercises. The student is required to have tried each exercise before being considered finished. The interactive character of the examples often leads to immediate natural feedback, such as "Did our simulated patient survive?" A minimum passing score could be specified for some topics as well. Failed exercises and principles can be used as a basis to retrieve similar examples for remedial action.

6.1.4   Multimedia

Because multimedia is used throughout the ITS to present both the course material and the examples, the multimedia will be represented in a multimedia object. This object will include information on the media's display window such as its location, size, buttons, capabilities, as well as the type of media and the protocol to display and run it. At a minimum, the following media will be supported.

- Interactive Simulations
- Interactive Animations
- Video
- Interactive Photos
- Hypertext
- Audio

Existing multimedia software will be used whenever possible.

Interactive simulations require that the GITSDC include a simulation development capability as well as facilities to run existing simulations. The simulation development capability consists of basic simulation objects and their associated methods. A set of simulation primitives which describe how objects interact and are updated will be included. A means will be provided to allow the developer to specify what information may be input from the student, what defaults to use, and what information must be output to the student. The capabilities to run existing simulations will include the inputs needed to the simulation, the calling convention, and the output format. Various calling conventions will be supported including DDE, DDL, and Executables.

An animation development facility will be developed for the GITSDC to allow interactive animations. A set of visual objects will exist with an easily defined form, such as a user definable bitmap. These objects can be attached to simulated objects for their position and visual mode data.

Currently, several video drivers exist which can display video clips on a variety of platforms from a variety of formats in a variety of resolutions. These drivers will be researched and the most appropriate one utilized. The type of interaction with the video clips must be specified in a particular ITS. The GITSDC must accommodate this specification. One method is to allow the easy placement of buttons on the video window with attached methods. Specification of video branching options must also be allowed through the representation of possible video branches and the method of selecting between them.

Currently, software exists which displays high and low resolution photographs, bitmaps, and other graphic information in a variety of formats. These allow user zooming and panning. The most appropriate software will be selected and utilized. Furthermore, the GITSDC must allow the specification of mouse sensitive regions along with the appropriate action to take in response to mouse clicks in the region. Possibilities include displaying another media object of any type or zooming into that area.

Several hypertext development programs exist. These will be investigated and if one is suitable it will be selected and used. Otherwise, this capability can be developed for the GITSDC relatively easily. Facilities must exist to define mouseable areas or terms in the text and pictures and connect those with other media objects or actions.

Currently, drivers exist to play audio clips in a variety of resolutions from a variety of formats. These will be investigated and the most appropriate one included in the GITSDC. The type of interaction with the audio clips must be specified in a particular ITS. The GITSDC must accommodate this specification. One method is to allow the easy placement of buttons on the video window with attached methods. Specification of audio branching options must also be allowed through the representation of possible audio branches and the method of selecting between them.

6.1.5   Student Model

In order to tailor the course of study to the individual student, we will keep an explicit model of each student using the ITS. The student model will include a list of what Topics the student has examined and what examples he has experienced along with the date and time

that interaction occurred. A complete history of his performance on all exercises is maintained. Based on the pattern of his unsatisfactory performance on exercises, a set of topics and principles, or combinations of them, can be developed which form a hypothesis as to what information the student does not understand. Based on this hypothesis and the exercises missed, similar examples can be shown to increase the student's understanding. It may be appropriate to force the student to re-experience some of the Topics or to show the student more detailed information on a specific topic as well. Based on the hypothesis, a new set of exercises can be generated for testing the success of the remediation.

6.2 Customizations

In addition to the general ITS development capability, the ITS will also cover topics for different sections of the course. The course developers for the 91 B course will select the parts of the courses which they would like to cover in the ITS(s) developed. Input from instructors for the content of the ITS will be very important to the success of the ITS development. Review of the material in the ITS by the appropriate specialists will also be extremely important so that the ITS shows the correct procedure. Incorrect information or missing crucial steps or precautions of a procedure could lead to life-threatening problems if followed by a student.

In Phase I we developed a prototype ITS for critical tasks such as intubation, needle chest decompression, and manual defibrillation. These tasks could be expanded into a full-scale ITS which includes all similar critical psychomotor skills from the 91 B level 2 course. Optionally, separate individual ITSs could be developed for separate tasks. The simulation developed for the prototype could be refined for the current tasks and expanded to include additional tasks.